Proceedings of the Institute of Acoustics

MAXIMUM LIKELIHOOD AND MAXIMUM MUTUAL INFORMATION TRAINING OF CONTINUOUS DENSITY HIDDEN MARKOV MODELS - EXPERIMENTS ON THE E-SET

S.Kapadia, V.Valtchev, S.J.Young

Cambridge University Engineering Department
Trumpington Street, Cambridge, England.

1. INTRODUCTION

Traditionally Hidden Markov Models (HMMs) are optimised according to the Maximum Likelihood (ML) criterion using the Baum-Welch (BW) algorithm [8]. Whilst the ML estimation of model parameters maximises the likelihood of the speech data given the model, it does not optimise a meaningful decision theoretic criterion such as the expected error rate. Consequently the ML approach fails to give good parameter estimation for limited amounts of training data even when the model space includes the true source. On the other hand, when the family of parametric distributions described by the model includes the distribution of the source and there is an infinite amount of training data available, then the global ML estimate is optimal in the sense that it yields an unbiased estimate with minimum variance. Unfortunately, when constructing HMM-based speech recognisers training data is not unlimited and the model space includes no member that even resembles the true distribution of the source. In these cases, examples can be constructed [4] where the Maximum Mutual Information (MMI) estimator can produce better models than the corresponding ML estimator [1].

All of these theoretical advantages of MMI over ML are well known. However, clear demonstrations of the practical utility of MMI when applied to large speech recognition tasks remain elusive. MMI training involves a number of practical difficulties. The Baum-Welch (BW) algorithm is a fast and efficient algorithm for ML parameter estimation. Unfortunately in its extension to MMI [3], the practical implementation of the algorithm looses the guarantee of improving the objective function.

Consequently, due to the lack of theoretical guidance, past research on MMI has tended to use somewhat slow and unreliable gradient descent methods. We have begun a systematic empirical study of the various methods used to implement MMI training for HMM parameter estimation. This paper reports the progress of our preliminary work which investigates the applicability of fast first order derivative methods to MMI training. A set of speaker independent recognition experiments are presented on a 104 speaker British English E-set database. Conventional HMMs with multiple mixture continuous output distributions are trained for each member of the E-set. Results are presented for both diagonal and full covariance HMMs estimated using Maximum Likelihood (ML) and Maximum Mutual Information (MMI) training criteria. The ML training is carried out using the standard Baum-Welch reestimation algorithm. The MMI training utilises Scott Fahlman's QuickProp which exhibits extremely fast and stable convergence. Virtually all results show clear performance gains achieved by MMI training and are comparable to the best reported by other researchers.

2. ML AND MMI OPTIMISATION OF HMM PARAMETERS

In the ML estimation approach, given an acoustic observation $y(n)$ and associated transcription $tr(n)$ where $n = 1...N$ the parameter set λ is estimated so as to maximise

$$L_\lambda = \sum_n \log p_\lambda(y(n)|tr(n)) \tag{1}$$

where $p_\lambda(y(n)|tr(n))$ is the probability of the acoustic observations from an HMM with parameters λ built to the transcription $t(n)$. The Baum-Welch algorithm which is most commonly used for this task applies a transformation on the parameter set λ which is guaranteed to converge on a local maximum of L_λ. For example, the transition probabilities $a_{i,j}$ are re-estimated using the formula

$$a_{i,j}(t) = \frac{a_{i,j}(t-1)\frac{\partial L_\lambda}{\partial a_{i,j}(t-1)}}{\sum_{k=1}^{N} a_{i,k}(t-1)\frac{\partial L_\lambda}{\partial a_{i,k}(t-1)}} \tag{2}$$

In the MMI approach the parameters of the model are reestimated by maximising

$$I_\lambda = \sum_n \log p_\lambda(y(n)|t(n)) - \log p_\lambda(y(n)|r) \tag{3}$$

where r represents the recognition-time HMM. In our case, the recognition model adopts the structure shown in figure 1 which is the composite system of word models including any language model.

3. OPTIMISATION OF THE MMI OBJECTIVE FUNCTION

Traditionally MMI optimisation of HMM parameters is carried out using some form of gradient ascent. The partial derivative of the cost function is calculated with respect to each parameter in the system and, using this information, gradient ascent is performed in the parameter space. The method is guaranteed to converge onto a local maximum only for infinitesimal steps taken in the direction of the gradient. For example, the update equation of an *unconstrained* transition parameter $a_{i,j}$ would be

$$a_{i,j}(t) = a_{i,j}(t-1) + \eta\frac{\partial I_\lambda}{\partial a_{i,j}(t-1)} \tag{4}$$

where η is the step size. In order to reduce the time needed to find a solution, it is desirable to take the largest possible step without overshooting the solution. Unfortunately, the set of partial derivatives computed at a single point in the parameter space does not contain enough information to do this safely. One way to tackle this problem would be to dynamically adjust the step size depending on previously computed derivatives. A form of this strategy is to use a momentum term which adds a small amount of the previous change to the current update

$$a_{i,j}(t) = a_{i,j}(t-1) + \Delta a_{i,j}(t) \tag{5}$$

$$\Delta a_{i,j}(t) = \eta\frac{\partial I_\lambda}{\partial a_{i,j}(t-1)} + \zeta\Delta a_{i,j}(t-1) \tag{6}$$

In very complex systems, it is often advantageous to have a separate step size for each parameter. Jacobs [5] has conducted an empirical study comparing standard *Back Propagation* of the above form to an update rule that dynamically adjusts a separate step-size parameter for each weight in a neural network. The same rule has successfully been used by Robinson [9] for the training of his recurrent error-propagation network.

Another approach to improving the speed of convergence is to make explicit use of higher order derivatives [6]. Let $\mathbf{a}(t-1)$ be the vector containing the present values of all parameters in a system. Given higher order derivative information, a new parameter vector $\mathbf{a}(t)$ can be computed by

$$\mathbf{a}(t) = \mathbf{a}(t-1) - \eta \mathbf{H}^{-1}\mathbf{g}(t) \tag{7}$$

where $\mathbf{a}(t-1)$ is the old parameter vector, $\mathbf{g}(t)$ is the gradient of the objective function with respect to the parameter vector and $\mathbf{H}$ is the Hessian matrix of second derivatives. The full Hessian of a system with n parameters will have n^2 elements. In order to reduce the computational load due to the calculation, inversion and storage of the Hessian matrix most implementations of this method use some approximation to the Hessian. The rate of convergence then depends on the accuracy of this approximation. In its simplest form such approximations use prior information to zero parts of the Hessian matrix. For example, the assumption that different sets of parameters are independent will result in a block-diagonal Hessian matrix. In the work presented here, we adopt the rather gross assumption that all parameters in an HMM are independent. We further simplify the computation by using a difference approximation to the second derivatives rather than exact values.

$$\mathbf{H} = [h_{ii}] \tag{8}$$

$$h_{ii} = \frac{\partial^2 I_\lambda}{\partial a_i^2} \approx \frac{\frac{\partial I_\lambda}{\partial a_i}(t) - \frac{\partial I_\lambda}{\partial a_i}(t-1)}{\Delta a_i(t-1)} \tag{9}$$

Using equations 9 and 7 gives

$$\Delta a_i(t) = -\eta \frac{1}{h_{ii}} g_i(t) \tag{10}$$

$$= \eta \frac{g_i(t)}{g_i(t-1) - g_i(t)} \Delta a_i(t-1) \tag{11}$$

If η in the above equation is chosen to be 1.0, the equation becomes identical to the update strategy of QuickProp proposed by Fahlman [2]. Although the value of $\Delta a_i(t)$ is only an approximation to the optimal change in parameter, we have found this method to be very effective when applied iteratively.

The behaviour of the update rule given by equation 11 with $\eta = 1.0$ is as follows. If the current gradient is smaller than the previous one but in the same direction, the parameter will change again in the same direction. The step taken may be large or small depending on how much the gradient was reduced by the previous step. If the current slope is in the opposite direction from the previous one, then we have stepped beyond the maximum. In this case, the next step will place us somewhere between the current and the previous position. The third case occurs when the current gradient is in the same direction as the previous but is of the same size or larger in magnitude. If we were to blindly follow the formula we would end up taking an infinite step or moving in the wrong direction. The third case occurs naturally since the update rule given by equation 7 will converge

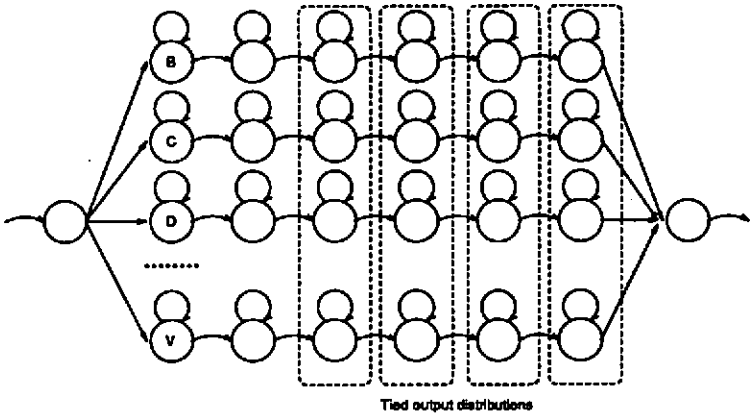

Figure 1: Model structure for E-Set Recognition

to the nearest turning point. In order to handle this special case, we adopt the method used by Fahlman in QuickProp. No parameter change is allowed to be greater in magnitude than μ times the previous update for that parameter. If the change computed by the update formula is too large, infinite or in the opposite direction to the current gradient, we instead use μ times the previous change as the current change. The optimal value of μ is problem specific and we have chosen the value of 1.75 for our present experiments. In his neural net experiments, Fahlman observed that if μ is too large the system behaves chaotically and fails to converge.

A bootstrap process is also used to provide initial values of the parameter changes. More generally, if the previous gradient is zero or non-existent then the current change in parameter is calculated using plain gradient ascent with some learning rate $\tilde{\eta}$.

4. EXPERIMENTS & RESULTS

The task chosen to evaluate the performance of the two training techniques was the speaker independent (SI) recognition of the members of the British English E-set ("B", "C", "D", "E", "G", "P", "T" & "V"). E-set recognition is considered to be a particularly difficult task due to the high level of confusability between the different classes in the set. The data used for the experiments was collected and distributed by British Telecom Laboratories and forms a subset of their spoken alphabet database. Each member of the E-set is represented by three utterances from each of the 104 different speakers (54 males, 50 females). The speakers are split into two halves to form a training set of 1239 utterances and a test set of 1219 utterances. The acoustic preprocessor used the output of a 27 channel filterbank followed by a Discrete Cosine Transform to produce 12 Mel Frequency cepstral coefficients (MFCCs) and their first order differentials (the twelve coefficients include the zeroeth coefficient which is the average value of the log power spectrum). The preprocessor and

Type	*log(P)*	*Train %*	*Test %*
1/Diag	−30.385	91.04	84.50
1/Full	−27.681	98.87	92.29
3/Diag	−28.660	95.24	88.76
4/Diag	−28.354	96.85	90.65
5/Diag	−27.892	96.77	90.40
8/Diag	−26.983	98.87	90.40

Table 1: Maximum Likelihood Results (SI) for 15-state HMMs, states 7-15 tied. The first column gives the number of mixtures and type of covariance matrix.

Type	*Cross-entropy*	*Train %*	*Test %*
1/Diag	$-7.7866e^{-07}$	100.00	90.40
1/Full	$-6.9866e^{-07}$	100.00	93.68
3/Diag	$-1.3400e^{-09}$	100.00	90.81
4/Diag	$-5.0000e^{-07}$	100.00	90.81
5/Diag	$-4.2686e^{-06}$	100.00	89.75

Table 2: Maximum Mutual Information Results (SI) for 15-state HMMs, states 7-15 tied. The first column gives the number of mixtures and type of covariance matrix.

the partitioning of the data are, in fact, absolutely identical to the ones used by Woodland in [10]. All training and testing used the HTK Portable Toolkit plus extensions for MMI training [11].

All HMMs were strictly left to right with no skips. Eight iterations of ML estimation were used to produce the models with performance given in Table 1 (all parameters updated). Twelve to twenty five iterations of QuickProp were then used to produce the MMI models whose performance is given in Table 2(means only). In both of these cases, states 7-15 of all the models were tied to give common modelling of the vowel portion of the E-set words.

In the MMI training, the conditional cross entropy always increased from iteration to iteration and it is well within the neighbourhood of the theoretical maximum after the first 10 - 15 iterations (see figure 4.). This represents fast and stable convergence which is far superior to standard first order gradient ascent methods.

In all experiments the MMI-trained models achieved 100% recognition on the training set which suggests the need for larger training databases. For comparison Woodland [10] reported 0.5% (train set), 7.9% (test set) using HMMs with discriminative output distributions and reduced feature vector size.

Table 3 demonstrates the effect of sharing parameters amongst models [12]. The first column indicates the number of mixtures and the kind of covariance of the matrix. For the latter, *Diag* denotes diagonal covariance; *Full* denotes full covariance; *GCov* denotes a single grand covariance tied across all models; and *GMCov* denotes a single grand covariance for each individual model. A clear improvement in performance is visible when the vowel states share the same output dis-

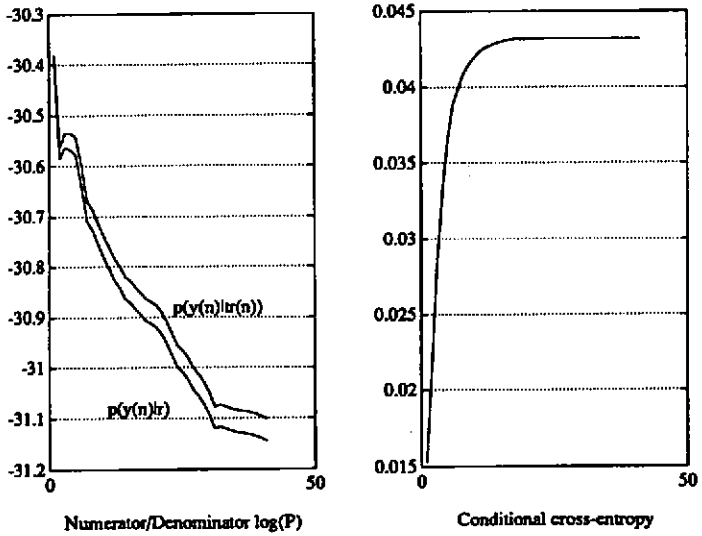

Figure 2: The likelihood functions and conditional cross-entropy plot for *1mix/Diag* MMI training. The maximum theoretical value of the average frame conditional cross entropy can be computed by pn/m where p is the entry model log probability, n is the number of training utterances and m is the total number of frames. For $p = \log 1/8$, $n = 1239$ and $m = 59644$ the maximum theoretical cross entropy is 0.043196.

Type	*Tying*	*log(P)*	*ML train %*	*ML test %*	*MMI test %*	*θ*
1/Diag	*no tying*		90.88	81.21	84.74	5760
1/Diag	7-15	−30.385	91.04	84.50	90.40	2736
1/Full	7-15	−27.681	98.87	92.29	93.68	19152
1/GCov	7-15	−32.685	88.70	84.09	89.17	1704
1/GMCov	7-15	−29.836	95.00	90.65	91.63	4056
3/Diag	*no tying*			80.64		17280
3/Diag	7-15	−28.660	95.24	88.76	90.81	8208
3/Diag	5-15	−28.891		88.52	89.01	6192
4/Diag	7-15	−28.354	96.85	90.65	90.81	10944
4/Diag	6-15				91.39	9600
4/Diag	5-15				91.71	8256

Table 3: The effect of tying various parameters (SI). The first column describes the number and type of mixtures used in the state output distributions (see text). The second column shows which states have their output distributions tied across all models. λ is the total number of parameters used in the models excluding transition probabilities.

tributions across all models. This is due to the fact that virtually all of the information needed to discriminate between the different classes is concentrated in the consonant part of each utterance. Having separate vowel models (possibly not so well trained due to the small amount of data) will increase the chance of confusion occurring between the classes in the final stages of the Viterbi search. The table also shows that full covariance modelling provides greater accuracy than diagonal covariance and that tying a full covariance matrix across all the states of a single model provides a very compact but effective system.

5. CONCLUSIONS & FUTURE WORK

This paper has presented a new implementation of the MMI training algorithm using the QuickProp update strategy. QuickProp was originally derived empirically, here we have shown that although it looks like a first order method, it can be regarded as a classical second order optimisation technique which uses a crude (but effective) approximation to the Hessian matrix.

Results on speaker independent E-set recognition have been presented which show that MMI training substantially improves recognition performance, but the improvement relative to the corresponding ML case decreases as the model complexity increases. The best score of 93.6% correct is comparable to the best results published elsewhere on this task and the uniform 100% performance achieved on the training data suggests strongly, that much better test performance could be achieved if more training data was available.

Future research will concentrate in two main areas. Firstly, the applicability of MMI training to large continuous speech recognition tasks will be investigated. Secondly, further second order training algorithms will be studied in order to find computationally efficient mechanisms for exploiting MMI and other similar discriminative techniques in large-scale tasks.

REFERENCES

[1] P.F. Brown. The Acoustic-Modeling Problem in Automatic Speech Recognition. Technical report, IBM Thomas J. Watson Research Center, August 1987.

[2] S.E. Fahlman. An Empirical Study of Learning Speed in Back-Propagation Networks. Technical report, CMU, September 1988.

[3] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo. An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems. *IEEE Transactions on Information Theory*, 37(1), January 1991.

[4] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, D. Nahamoo, and M.A. Picheny. Decoder Selection Based on Cross-Entropies. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1988.

[5] R.A. Jacobs. Increased Rates of Convergence Through Learning Rate Adaptation. *Neural Networks*, 1, 1988.

[6] D. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Massachusetts, 1984.

[7] A. Nádas. Optimal Solution of a Training Problem in Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(1), February 1985.

[8] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), February 1989.

[9] A.J. Robinson. *Dynamic Error Propagation Networks.* PhD thesis, Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2, June 1989.

[10] P.C. Woodland and D.R. Cole. Optimising Hidden Markov Models using Discriminative Output Distributions. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing.* IEEE, April 1991.

[11] S.J. Young. *HTK: Hidden Markov Model Toolkit V1.4 - Reference Manual.* Cambridge University Engineering Department, September 1992.

[12] S.J. Young. The General Use of Tying in Phoneme-Based HMM Speech Recognisers. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing.* IEEE, March 1992.